REMARKS

Claims 2-5, 7-19, 21-28, 30, and 32-34 are pending in the case. Claims 1, 6, 20, 29, and 31, previously canceled in light of allowability indicated by the Examiner, have been re-presented as claims 42, 43, 44, 45, 46, respectively, and the dependencies have been adjusted accordingly. Additionally, claims 7, 21, and 30 have been amended to restore them to their condition before claims 1, 6, 20, 29, and 31 were canceled. Claims 35-41 have been added. Claims 2-5, 7-19, 21-28, 30, and 32-34 are rejected in the current office action. Also, the drawings are objected to by the Examiner and six references (Mathis, Fogg, Newton, White, Wilkinson, Hagemann) have been newly cited in the case.

In particular and in accordance with the item numbering of the office action:

In item 3, claims 21, 26, 28, 30 and 34 are rejected as being unpatentable, under section 103(a), over Sonesh (U.S. 6,046,762) in view of Mathis (U.S. 6,269,254);

In item 4, claims 2-5, 7-15, 17 and 23-25 are rejected as being unpatentable, under section 103(a), over Sonesh in view of Mathis and Fogg (U.S. 5,841,839);

In item 5, claim 16 is rejected as being unpatentable, under section 103(a), over Sonesh, Mathis and Fogg, in view of Newton;

In item 6, claim 19 is rejected as unpatentable, under section 103(a), over Sonesh, Mathis, and Fogg, in view of White (U.S. 6,438,559);

In item 7, claims 22 and 18 are rejected as unpatentable, under section 103(a), over Sonesh, Mathis, and Fogg, in view of Wilkinson (U.S. 6,492,989);

In item 8, claim 27 is rejected as unpatentable, under section 103(a), over Sonesh and Mathis, in view of Fogg;

In item 9, claim 32 is rejected as unpatentable under section 103(a), over Sonesh and Mathis in view of Hagemann (U.S. 6,577,724); and

In item 10, claim 33 is rejected as unpatentable under section 103(a), over Sonesh and Mathis in view of White (U.S. 6,438,559).

In addition the Examiner has admitted that the combination of Sonesh and Reksten does not teach the call object of applicant's invention.

In response, Applicant repeats all of his earlier arguments including those in the office action response filed on July 23, 2003. Applicant also provides the following remarks.

Regarding Item 3, the Examiner contends that the combination of Sonesh and Mathis teaches or suggests Applicant's invention as recited in claims 44 (formerly claim 20) and 21. Applicant respectfully submits that the combination of Sonesh and Mathis does not teach or suggest Applicant's invention.

Sonesh describes a multimedia telecommunication automatic call distribution system. The system of Sonesh is said to provide a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means, including simultaneous voice, data, and video telephony, and at the same time ensures effective transparent spreading of agents over different geographical locations. Sonesh, Col. 3, lines 62-67. The invention of Sonesh uses a multimedia automatic call distribution (MMACD) server acting as a connection manager for data network callers and provides for automatic caller identification. Sonesh, Col. 3, line 67 to Col. 4, line 3.

The Mathis reference describes a modification of the general JTAPI model to accommodate a radio communications device. According to the Mathis specification, existing JTAPI does not support a dual-mode call and requires too large a memory allocation, because it requires over 63[1] event classes. The Mathis invention solves the first of these problems by permitting a method, call.connect( ) to be overridden[2] to add a method that takes as a first argument a terminal array, instead of a single originating terminal. Mathis, Col. 6, lines 7-11. The

[1] As explained in the Mathis specification the JTAPI definition has a separate class for each different JTAPI event. This burdens the mobile station with defining over 63 event classes, with a total class file of approximately 130 Kbytes. Mathis, Col. 8, lines 54-57.

terminal array is the argument of the method and contains an array of terminal objects which may be a voice terminal and a data terminal or a voice terminal and a fax terminal, or a data terminal and a fax terminal. The second problem is apparently solved in Mathis by replacing a multitude of event classes with a smaller set of event-category classes. Eight generic classes are provided, in Mathis, and specific events are grouped into these generic classes according to Table 3 of the reference. Applications use an event ID to determine a specific event within a broad type. Mathis, Col. 8, lines 58-65.

As further explained in Mathis, in the Appendix of the specification, JTAPI is a portable, object-oriented application programming interface for Java-based computer telephony applications. Mathis, Col. 10, lines 26-34. The Java Telephony Call Model includes a half-dozen Java objects wherein each call model object represents either a physical or logical entity in the telephone world. FIG. 10 of Mathis illustrates the model. Included in the model are a Provider object, a Call object, an Address object, a Connection object, a Terminal object, and a Terminal Connection object.

In the Mathis invention as shown in FIG. 4, when a call is to be setup, a provider method creates a Call object which then creates a local Connection object. The local Connection object then creates, for a simple voice connection, a Terminal Connection object. Thus, Mathis uses, with some changes to the call.connect( ) method, the basic model of JTAPI.

Thus, the combination of Sonesh and Mathis, presumably, would have implemented the multimedia automated call distribution system of Sonesh by means of the JTAPI of Mathis. However, an automated call distribution system implemented using JTAPI is not Applicant's invention. Applicant's call manager object and call object (despite the similarity in name) are not identifiable, and should not be identified, with the objects in JTAPI. The objects in Applicant's invention do not implement the JTAPI call model in which there is an object for each physical or logical part of the telephone world. Applicant's invention has only two object classes, a call

[2] If a method is declared with the same name and parameters as a method in a superclass, the method in the superclass is considered to be overridden. A superclass is a more generic class that includes other classes that are more specific.

manager object and a call object in the Framework layer[3], which resides on a JavaISDN layer, which in turn resides on a WinISDN layer, the latter being the software driver for the hardware. Applicant's specification, FIG. 4. The WinISDN driver manages the ISDN adapter hardware. The JavaISDN layer acts as an interface between the WinISDN layer and the Framework layer in which the call manager object and the call object reside. This is not the structure of JTAPI and there is no intent to be similar to that structure. Also, Applicant's invention does not create or use any other objects, as does the JTAPI system and Applicant's call manager and call objects are not directed implementing a model of each part of the telephone world. Applicant's objects are high level objects that interoperate with the JavaISDN layer to perform their functions. Finally, Applicant's call object does not have overridden methods because such an object presumes a predefined set of classes (such as those in JTAPI) whose methods are inherited by subclasses. Applicant's invention presumes no such predefined set of classes. Thus, the proposed combination does not result in Applicant's invention as recited in claims 44 and 21.

The Office Action has alleged that the ACD minicomputer of Sonesh is the call manager object because it is the software on the ACD-minicomputer that processes the calls. Applicant submits that the ACD minicomputer cannot be identified as a call manager object. In fact, such an association is at odds with the Examiner's use of Mathis. If the Examiner believes that the ACD-minicomputer is an object in the sense of the word used in Mathis, the Examiner should identify the ACD-minicomputer with one of the JTAPI objects in Mathis. No such identification has been made. Applicant is left wondering whether the Examiner believe that the ACD-minicomputer is a different kind of object from those disclosed in Mathis.

The Examiner has also alleged that playing a voice menu describing a plurality of selection items in a department table, as recited in claims 44 and 21, is found in Sonesh. However, Sonesh does not mention any such table. Sonesh only states that there is a routing algorithm and never mentions anything about a department table. Sonesh, Col. 5, lines 44-50. The Examiner admits as much by quoting the term department table in the Office Action,[4] which

[3] "Having a Framework layer comprising only two basic, comparatively simple objects, hides the actual complexity of the system but makes the system easier to deal with at the application layer." Applicant's specification, page 7, lines 31 to page 8, line 1.

[4] "The examiner maintains the "department table" of Sonesh is accessed by the user selecting the desired criteria." Paper 11, Office Action, pages 3-4.

Applicant takes to mean that the Examiner has not really found, in Sonesh, the department table in the present invention. In fact, a close examination of the Sonesh reference reveals that the caller never really has direct control, as he does in the present invention, over the routing of his call. The caller, in Sonesh, instead indicates his area of interest or service and the MMACD then routes the call based on this input and the caller identification information previously gathered. Sonesh, Col. 7, lines 17-24. This is distinctly different from the call self-routing function of the present invention, which uses department tables that correspond to the departments of an organization. A caller in Applicant's invention routes himself by his selections through possibly a sequence of departments of the organization.

The Office Action concludes that "it would have been obvious to one of ordinary skill in the art, having both Sonesh and Mathis before him, to be motivated to modify the system of Sonesh by storing the user's information in a Java object." Office Action, page 4. This motivation, it is argued, derives from a desire to "improve the system, since Java code is easily transported across different computer platforms."

Applicant responds that the combination of Sonesh and Mathis, if made, would have resulted in the MMACD system of Sonesh being implemented with JTAPI and the modifications[5] that Mathis makes to JTAPI. Again, as explained above, this is quite clearly not Applicant's invention, as recited in claims 44 and 21. Furthermore, Applicant's motivation is not a desire to make a transportable call distribution system. Applicant's motive, with respect to the use of only the call manager object and the call object classes is to provide a simplified framework[6] and to make the system easier to deal with at the application layer. Applicant's specification, page 7, line 31 to page 8, line 1. Regarding the use of department tables, Applicant's motivation is to relieve the burden on the application designer, such that no programs need to be written to have a usable application. By simply setting up the department tables and voice menus according to the structure of the organization, a usable application results. Applicant's specification, page 11, lines 10-14. Therefore, the motivation suggested by

[5] specifically, the overriding of the methods of the call.connect( ) method

[6] Quite clearly, the JTAPI system with 63 event classes, as described in Mathis, is not a simple framework for creating a call application. If it were, Mathis would see no reason to modify it for use in a radio communications device.

the Examiner is inapplicable to Applicant's invention. The motivation suggested by the Examiner is only reflective of what would have resulted if Sonesh and Mathis were combined, i.e., a call distribution system implemented with JTAPI. In fact, there is no reason to make the combination of Sonesh and Mathis, because the Mathis reference, by itself, points out the use of JTAPI to make an automated call distribution system using the Call Center Package. Mathis, Col. 12, lines 55-60. Furthermore, Applicant's invention achieves an advantage that is not available to that achieved by the proposed combination of Sonesh and Mathis, which is the relieving the need for programming of the call application. All that is required to set up a call distribution program is the setting up of department tables that correspond to the departments of the particular organization in question. This further permits the call manager and call object framework to be present in a plurality of computing nodes, as recited in claims 44 and 21.

Thus, the combination of Sonesh and Mathis fails to teach at least the limitation "each call object including the department table with which the call is currently associated," as recited in claims 44 and 21. Therefore, the proposed combination fails to teach all of the limitations of Applicant's invention as recited in claims 44 and 21.[7] Furthermore, there is no clear and particular teaching or suggestion in Sonesh or Mathis to modify the combination to use a department table in a call object in accordance with the present invention.

Applicant further submits that the above arguments apply to claims 42 (formerly claim 1) 26, 28, 45 (formerly claim 29), and 46 (formerly claim 31). Because all of the Office Action rejections are based on, at least, the combination of Sonesh and Mathis, all of the Office Action rejections relating to the independent claims have failed to make a prima facie case of obviousness, by virtue of the above argument. Therefore, Applicant submits that these claims are unobvious over the cited art. Furthermore, according to *In re Fine*, if an independent claim is nonobvious under 35 U.S.C. §103, then any claim depending therefrom is nonobvious.[8] Therefore, all of the dependent claims are unobvious as well, at least for the reasons stated above, as well as for all of the reasons previously submitted by the Applicant.

[7] See MPEP 2143.03 All Claim Limitations Must Be Taught Or Suggested. *In re Royka*, 180 USPQ 580 (CCPA 1974).

[8] *In re Fine*, 5 USPQ2d 1596, 1600 (Fed. Cir. 1988).

NEW CLAIMS 35-41

Applicant presents new claims 35-41. Claim 35 includes some of the specific methods of the call manager object and call object as well as the software driver and interface layers. Claim 36 is directed to the recursive aspect of embedding a table in the call object. Claims 37 and 38 are directed to different types of call selection inputs, claim 39 is directed to the organization database, claim 40 is directed to a method in accordance with the present invention and claim 41 is directed to a framework in accordance with the present invention. Consideration of these new claims and reconsideration of the pending and previously presented claims is respectfully requested.

Authorization is hereby given to charge any fee deficiency or credit any overpayment to deposit account 50-2778.

Date: February 10, 2005

Respectfully submitted

Anthony B Diepenbrock

Anthony B. Diepenbrock III
Reg. No. 39,960

DECHERT LLP
Customer No. 37509
Telephone: 650.813.4800
Facsimile: 650.813.4848

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being forwarded via facsimile to Examiner Ron Abelson in Group No. 2666 at facsimile number 571.273.3764 and 703.872.9306 located at Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on February 10, 2005.

Date: February 10, 2005

Yolette Yturralde-Owen